SPHERICAL AND STOCHASTIC CO-CLUSTERING ALGORITHMS

A Thesis

Presented to

The Faculty of the Department of Computer Science

Sam Houston State University

In Partial Fulfillment

of the Requirements for the Degree of

Master of Science

---

by

Emrah Sariboz

May, 2019

SPHERICAL AND STOCHASTIC CO-CLUSTERING ALGORITHMS

by

Emrah Sariboz

_____

APPROVED:

Hyuk Cho, PhD
Thesis Director

Cihan Varol, PhD
Committee Member

Rabieh Khaled , PhD
Committee Member

John B. Pascarella, PhD
Dean, College of Science and Engineering Technology

DEDICATION

It's dedicated to my parents who always supported me along my education

ABSTRACT

Sariboz, Emrah , *Spherical and stochastic co-clustering algorithms*. Master of Science (Computing and Information Science), May, 2019, Sam Houston State University, Huntsville, Texas.

Clustering, without a doubt, is a dominating area in data mining and machine learning field. Due to the wide range of the necessity to clustering algorithms, it has many applications in real-life problems, ranging from bioinformatics to personalized information delivery. Feature characteristics of the newly generated data lead us to new approaches to explore the nature of it. General single-sided (i.e. one-way) clustering algorithms such as k-means algorithm clusters either rows or columns of the data matrix. Co-clustering algorithm clusters both the instances and features of the data matrix simultaneously and thus, it is more suitable to discover the pattern(s) hidden in both row and column dimensions.

Most existing co-clustering algorithms include inexplicit clustering steps for each dimension, separately. In this study, we developed two novel co-clustering algorithms, named as Spherical Co-clustering and Stochastic Co-clustering, which utilize the existing k-means framework, furthermore a specific data construction, and two specific data normalization was included as a pre-processing step. The co-clustering framework resembles one existing co-clustering algorithm, spectral co-clustering, as it first applies feature selection using singular value decomposition and utilizes one-way clustering to achieve co-clustering. Furthermore, we partially address a couple of practical well-known problem in clustering algorithm which include the cluster initialization, the degeneracy problem, a local minimum, and a nan (not-a-number) condition in a Kullback-Leibler divergence.

The correctness and efficiency of the two algorithms were validated with publicly available benchmark dataset in terms of monotonicity of objective function value change and clustering accuracy. To be specific, we compared the accuracy performance of Euclidean k-means, stochastic k-means, spherical k-means, stochastic co-clustering and spherical co-clustering algorithms.

KEY WORDS: Spherical, Clustering, Co-clustering algorithm, K-means algorithm, Bi-normalization, Stochastic co-clustering, Sinkhorn-Knopp Normalization, Kullback-Leibler Divergence

ACKNOWLEDGEMENTS

There have been many people who encouraged me during my rewarding master journey. It's time to show the appreciation to them.

First of all, my advisor, Dr. Cho, whom was a primary professor helped me to discover my passion in a hodgepodge Computer Science field. I got inspired and encouraged to work on the Data Mining and Machine learning field. I would like to thank for his guidance, support, and understanding during the journey. His expertise illuminated my career during his advisory.

I would like to thank Varol family for their continuous support starting from my undergraduate degree. I had a collaboration opportunity work with Dr. Cihan Varol during my undergraduate degree which ended up being my very first publication. Dr. Hacer Varol was a great person to work for. During her assistantship, I gained the most valuable insights from different teaching aspects to programming.

Last but not least, I would like to thank Dr. Khaled for being a great professor during my both undergraduate and graduate degree. Also, thank you for being great committee member. His valuable suggestions, and insightful comments improved the quality of the study.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

Introduction

Data mining targets to discover information, relationships, changes, errors, and statistically significant information and values within the data (MacQueen, 1967). It can be defined as the exploration of the relationships in the large data set and the relationships that can be used to make predictions about the given data. Data usually are collected in the form of matrices for Machine Learning (ML) and Data Mining(DM) applications. Extracting meaningful and useful information hidden in the collected data matrix is a major task in a machine learning and data mining. Different characteristics of data require new approaches to explore the hidden patterns, which is one of the most critical difficulties to explore nature as well as the organization of data (whistworks.com, 2015).

One of the decisive methods representing a large branch of DM and attracting a lot of attention is the unsupervised learning, specifically clustering. Clustering is a data analysis tool for grouping data into several homogeneous groups (Jain, 2010). Objects, called as instances, tuples, and records, are generally divided into a certain number of clusters. While objects with the same characteristics are in the same group, objects with different characteristics are included in different groups. Therefore, the purpose of clustering is to group similar objects into a coherent cluster, while different objects to separated clusters (Anna Huang, Similarity Measures). To find and interpret the connections among data points there is a need for data clustering algorithms in many disciplines therefore, it has usages from bioinformatics to personalized information delivery (Cho et al. 2014).

K-means clustering is unsupervised learning algorithm, which is applicable to unlabeled data (Wagstaff et al. , 2001). The main objective of a k-means clustering is to minimize the distance between each data point to closest cluster centroid. To find the closest centroid, it originally uses Euclidian distance as a distance metric. The k-means clustering is an iterative algorithm, where it iteratively assigns each data point closest cluster. It clusters the given data matrix either row wise or column wise (Zhao, 1970) thus, it is also known as one-way clustering.

On the other hand, co-clustering (i.e. block clustering) clusters both row and column of the given data matrix simultaneously, or vice versa. Unlike k-means clustering algorithms, it seeks for block structure of rows and column that are inter-related (Cho and Dhillon, 2007). Although it is desirable over one-way clustering algorithm, it uses complex algorithm behind the scene. To address the problem, we developed two novel approaches to cluster given dataset in an efficient manner. To the best of our knowledge, spherical co-clustering and stochastic co-clustering have not been implemented by anyone yet. In this study, we apply both algorithms to two publicly available datasets: iris and seeds, to answer the following questions:

(1) Whether it is possible to cluster both row and feature of the data matrix simultaneously using bi-normalizing and Sinkhorn-Knopp techniques;

(2) Whether both stochastic and spherical co-clustering is desirable over tradition one-way clustering algorithms; and

(3) Whether it is possible to develop a co-clustering algorithm which uses traditional k-means clustering as a framework

Why co-clustering algorithm? In most cases, it is well known that co-clustering algorithms, a study of clustering of row and column of a given data matrix simultaneously, is desirable over a tradition one-way clustering algorithm from a number of perspectives:

Discovering Latent Structure: K-means algorithm, which clusters the given data matrix by row or column. Thus, it may fail to obtain the potential patterns (Cho, 2008). On the other hand, co-clustering algorithms clusters both instance and the feature of a given data matrix simultaneously, thus it seeks for block structure in a matrix. Therefore, it has more potential to discover the hidden pattern.

Dimensionality Reduction: One-way clustering algorithms can apply dimensionality reduction on only one side, either row or column; however, co-clustering can perform the dimensionality reduction at the same time (Cho and Dhillon, 2008). Co-Clustering algorithm can yield better quality clusters even when we apply one-way clustering since it shares column clustering information to row clustering information and vice versa.

As it can be understandable that the benefits of co-clustering make it desirable over traditional k-means clustering. However, current co-clustering algorithms uses complex ideas behind the scene. Due to crucial benefits over a one-way clustering algorithm, it has gained popularity across many fields, including, but limited to, gene expression, natural language processing, video, and product recommendation. Detailed review of the application areas of co-clustering algorithms can be found in the survey paper (Madeira and Oliveira, 2004). In this study, we develop two simple co-clustering algorithm that uses following specific strategies that can be directly applicable to k-

means framework: feature construction, data normalization, and a special cluster label initialization.

The remainder of the study organized as follows. In Chapter 2, we talk about the related works in clustering fields. We explain the main idea behind the works. In Chapter 3, we introduce a novel approach to the co-clustering field. In Chapter 4 and 5, we present the empirical results of the proposed algorithms and compare them to traditional one-way clustering algorithms.

CHAPTER II

Related Works

Clustering is an unsupervised algorithm where it groups similar objects together as clusters. The k-means (Hartigan, 1967) starts with initial cluster centers and assigns the cluster association for each data point to the closest centroid based on the Euclidian distance. Unlike one-way clustering that looks for similarity between rows or columns, co-clustering searches for "blocks" (or "common sets") of interconnected rows and columns (Hartigan, 1967). Co-clustering is an unsupervised data mining algorithm where its clusters both columns and the rows of the data matrix simultaneously to discover the “latent structures”.

The concept of similarity is not a clear information that can yield definitive results. Clustering algorithms require an accurate definition of the closeness between data points and cluster centroids, either the pair-wise similarity or distance (Huang, 2008). Meanwhile, the similarity is often considered in the aspect of similarity and dissimilarity. Although a variety of similarity of distance metrics have been proposed and adopted widely, in this section, we will briefly explain Euclidean distance and Cosine similarity. We also include the Kullback-Leibler divergence, which has been efficiently used in information theory-based clustering (Dhillon et al. , 2003).

K-means Clustering Algorithm

The co-clustering and k-means clustering algorithms have been developed parallelly. Both algorithms are unsupervised and serve clustering purposes. The core k-means algorithm has been implemented and reported and can be branched into two different application areas; single-pass, and parallel implementation (Forman & Zhang, 2000).

The metric on the k-means clustering algorithm is the usual Euclidian distance; which is used to measure the distance between each centroid and each data point. Firs we choose the K initial centroids, where K is a user specified parameter. Then each point is assigned to the closest centroid and assigned collection of points are now centroids. The centroid of each cluster is updated based on the number of points assigned to the cluster. We repeat this until we convergence, i.e. until the centroids are remain same.

Table 1

Pseudocode of k-Means clustering algorithm

---

1. Initialize ***row (or column)*** assignments and statics
2. ***repeat***
    2.1. Update ***all row (or column)*** assignments using Euclidean distance
    2.2. Update ***all column (or row)*** statistics
3. ***until*** convergence

K-means Metrics

The objective function, i.e., the cost function is a metrics that the applied algorithm wants to minimize or maximize according to the metric used. K-means algorithm uses the Euclidian distance as metrics to find the nearest cluster centroid to each point (Hartigan, 1979). In this case, we want to minimize the squared distance of each point to its closest centroid.

$$Q\left(\{\pi\}_{j=1}^{k}\right) = \sum_{j=1}^{k} \sum_{x \in \pi_j} \left\| x - C_j \right\|_2^2$$

where x is a data point that belongs to cluster (or partition) π_j and C_j is the cluster centroid of cluster j.

Note that both every data point and every centroid should be properly normalized for the new cluster assignment with each different metric. For example, Euclidean distance requires no data normalization; Cosine similarity requires L2-norm; and KL divergence requires L1-norm normalization. In what follows, we assume that both data points and centroids are properly normalized, unless otherwise mentioned.

Euclidean Distance

Traditional k-means clustering algorithm uses Euclidean distance as a metric to compute the distance between the data point and cluster center.

$$Distance_{x\,C_i} = \sqrt{\sum_{i=1}^{m}(x - C_i)^2}\,,$$

where x is data point and C_i is cluster centroid. The result of Euclidian distance ranges between 0 and positive infinity, where Euclidian 0 means both data points and cluster centroid are identical.

Cosine Similarity

Spherical K-means and spherical co-clustering algorithms uses cosine similarity as a metric.

$$\cos(x, C_i) = \frac{x^T C_i}{\|x\|\|C_i\|}$$

It defines either similarity or dissimilarity between data point and cluster centroid, where x is data point and C_i is cluster centroid. The result of Cosine similarity ranges between -1 to +1, -1 means the angle between the two vectors are 180 degree (i.e., complimentary); and 0 means 90 degree (i.e., orthogonal); and +1 means the angle between the two vectors is zero, therefore cosine value of 0 means vectors are identical.

Kullback-Leibler (KL) Divergence

In information theory-based clustering, KL measures the dissimilarity between two probability distribution (Kullback and Leibler, 1951) . Given two uncertain objects, in this case data point and cluster centroid, D (x || C_i) evaluates how probability distribution of x is differ from Ci,

$$D(x\,|C_i|) = -\sum P(x)\log\left(\frac{x}{C_i}\right),$$

Where both x (data point) and C_i (cluster centroid) are L1 norm normalized probability vectors. The result of KL divergence ranges between 0 and positive infinity. The result of zero means the two probability distributions are identical. The KL measures the divergence between the two-probability distribution (Thomas and Cover, 2006).

Monotonicity of Objective Function Value Change in Euclidean K-means

Objective function of the k-means clustering is Sum of Squared Error (SSE) between every data point and each closest centroid, where its objective is to minimize it (Tan et al., 2018).

$$\text{SSE} = \sum_{i=1}^{K}\sum_{X\in C_i}\|x - C_i\|_2^2 ,$$

where x = data point, C_i i'th cluster and K is number of clusters. The monotonicity of the SSE is proved in (Tan et al., 2018). Note that the first improvement of the objective function is obtained by the greedy cluster assignment ,and the second improvement is guaranteed by the process of computing the cluster centroid as follows:

$$\frac{\partial}{\partial_{c_k}}SSE = \frac{\partial}{\partial_{c_k}}\sum_{i=1}^{K}\sum_{x\in C_i}(x - C_i)^2$$

$$= \sum_{i=1}^{K}\sum_{x\in C_i}\frac{\partial}{\partial_{c_k}}(x - C_i)^2$$

$$= \sum_{x \in C_i} 2 \times (x - C_i)^2 = 0$$

Existing known open problems in clustering algorithm based upon the k-means framework.

Initial Cluster Assignment

K-means algorithm requires to have the initial cluster assignment. It is known that the result of the traditional k-means is sensitive to its initial cluster assignment. Traditionally, the following heuristics have been applied:

- **Random cluster label assignment.** Randomly assign a cluster membership to every row (or) column. In this case, each initial centroid should be computed by taking an average of all the data points that belong to the corresponding cluster.
- Random choice of k centroids from actual data. Randomly choose k data points and use them as initial k centroids. In this case, no initial centroid computation is needed as the randomly chosen k original (i.e., existing) data points are used as centroids.
- Random generation of k centroids. Randomly generate k initial centroids. In this case, no initial centroid computation is needed.
- Use of another clustering result as an initial seed. For example, the result of Hierarchical Agglomerative Clustering (HAC) is used as an initial cluster assignment (Cho and Dhillon, 2008). In this case, each initial centroid should be computed by taking an average as for the case of random assignment of the cluster membership to every row (or column).

Particularly, in this study, the co-clustering result from spectral co-clustering (Dhillon, 2001) is suggested to use the initial clustering seed. Details of the proposed initialization approach and its performance are discussed in the experimental chapter.

Update Cluster Assignment for Rows (or Columns)

Without loss of generality, we will discuss how to update the row cluster assignment. The column assignment can be applied in the same manner. For each row, first, compute the Euclidean distance to every cluster centroid and then update the row cluster assignment to the cluster membership (i.e., id) of the closest centroid.

Convergence Test

As the k-means algorithm (shown in Fig. 1) is an iterative algorithm, it will stop after some iteration. Conventionally, the following stopping criteria have been used.

- Fixed number of iteration (e.g., 100)
- Change of either absolute (i.e., $|Obj_{old} - Obj_{new}|$) or relative objective function (i.e. $|(Obj_{old} - Obj_{new})/ Obj_{old}|$) values between previous and current cluster assignments
- Change of centroids between previous and current cluster assignments

Note that the cluster centroids should be updated right after the row (or column) cluster assignment step. Otherwise, objective function value will not be changed. It is proven that the objective values are monotonically non-increasing over iterations.

Handling Empty Cluster Problem (i.e. Degeneracy Problem)

One of a famous problem with Euclidean k means clustering is empty clustering, which occurs if no data points are allocated for a cluster.

- Replace an empty cluster with a single instance, which is the farthest instance from its centroid (k-means++) (Arthur and Vassilvitskii, 2007)
- Generate a new centroid by assigning a new instance, which is randomly generated with a specific number (e.g., 1's).

Handling Not-a-Number in Computing KL Divergence

To avoid nan problem in KL divergence, we add a specific prior (i.e., epsilon) to the original data (data = data + epsilon, where epsilon ranges from 2.220446049250313e-16 to 100) as a data pre-processing step.

Spherical K-means Clustering Algorithm

Spherical k-means algorithm (Dhillon and Modha, 2001) brought a radically different perspective to clustering unlabeled documents. The intuition behind the algorithm is to remove the words that contain unique content from the document set and treat them as features and represent each document as a vector of certain weighted word frequencies in this property field. Finally, it uses the cosine similarity to find the similarity between vectors while as it is known, traditional k-means uses Euclidean distance to compare objects (Duda and Hart, 1973).

Table 4

Pseudocode of Spherical k-means Clustering Algorithm

1. Initialize row (or column) assignments and statics
2. ***repeat***
    2.1. Update ***all row (or column)*** assignments using ***Cosine Similarity***
    2.2. Update ***all row(or column)*** statistics
3. ***until*** convergence

As we have seen previously, the Euclidian distance is used as an objective function which we want to minimize it. As a major difference from the k-means clustering algorithm, spherical k-means clustering algorithm uses cosine similarity, i.e. inner product, to measure the similarity of the vectors (Dhillon and Modha, 2001):

$$x^T y = \|x\|\|y\|\cos(\theta),$$

where x and y are given two column vectors, $\|x\|$ is the vector 2-norm of x, and θ is an angle between x and y. Therefore, the overall objective function for the spherical k-means algorithm is defined as follows:

$$Q\left(\{\pi\}_{j=1}^{k}\right) = \sum_{j=1}^{k} \sum_{x \in \pi_j} x^T C_j$$

Co-Clustering

As we have stated earlier, co-clustering has a wide spectrum of usage from gene expression to document clustering. In this section, we will give a brief survey of previous work on co-clustering algorithms.

The initial idea of clustering both rows and columns were conceived under the name of "direct clustering" as a greedy algorithm which uses separation procedure to define the hierarchical block of columns and rows (Hartigan, 1967). One of the earliest application areas of the co-clustering has been used to cluster gene expression data analysis. The bi-clustering algorithm proposed by Cheng and Church was applied to cluster gene expression data and it advocated the importance of simultaneous clustering of genes and conditions for exploring more consistent and meaningful clusters (Cheng and Church, 2004).

Based on the squared residue formulated by the Cheng and Church (Cheng and Church, 2004), Minimum Sum-Squared Residue Co-clustering (MSSRCC) is introduced by Cho et al. to cluster genes and conditions. They have implemented two effective k means like a clustering algorithm to cluster the gene expressions. MSSRCC discovers the k rows and l columns of the given matrix while decreasing the monotonic order of square frames using bi-normalization and deterministic spectral initialization techniques.

The co-clustering algorithm was beneficial in grouping genes with similar functions under various conditions. Cho et al. proposed strategies to enhance the performance of MSSRCC algorithm. The enhanced MSSRCC algorithm has been used to the simultaneous exploration of the correlated samples of both genes and subsets. The algorithm has been applied to four publicly available microarray datasets. The accuracy measurement has been used to evaluate each subset (Cho et al., 2008). Co-clustering has been used in many areas including, but not limited to text mining, speech and video analysis and natural language processing (NLP).

The objective function of the MSSRCC algorithm is a 2-norm difference between A and Â;

$$\left\|\mathrm{A}-\hat{\mathrm{A}}\right\|_2^2,$$

where

$$\hat{\mathrm{A}} = RR^TACC^T,$$

where A is the input matrix, R is the row indicator matrix and C is column indicator matrix as explained below. This example adopted from Cho, 2008.

Given a data matrix, A

$$\mathrm{A} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & 0 & 0 \\ \mathbf{7} & \mathbf{8} & 0 & 0 \\ 0 & 0 & 4 & 5 \\ 0 & 0 & 6 & 7 \end{bmatrix}$$

We define a row cluster indicator matrix, $R \in R^{m\times k}$ and a column cluster indicator matrix, $C \in R^{n\times l}$ as follows: column r of R has m_r non-zeros, each of which equals m $\sqrt{m_r}$, the non-zeros of C are defined similarly. For is example, we assume that the first two rows (i.e., rows 1 and 2) belong to row cluster 1 and the remaining two rows (i.e., rows 3 and 4) belong to row cluster 2. Similarly, the first two columns and the remaining two columns of A belong to column cluster 1 and column cluster 2, respectively.

$$R = \begin{bmatrix} 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} \\ 0 & 1/\sqrt{2} \end{bmatrix}$$

$$C = \begin{bmatrix} 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} \\ 0 & 1/\sqrt{2} \end{bmatrix}$$

Compressed row cluster average and expanded row cluster average can be computed respectively by $B_r = R^T A$ (proper normalization necessary to get the row centroids) and $C_r = RR^T A$ as follows.

$$B_r = \begin{bmatrix} 4 & 5 & 0 & 0 \\ 0 & 0 & 5 & 6 \end{bmatrix}$$

$$C_r = \begin{bmatrix} 4 & 5 & 0 & 0 \\ 4 & 5 & 0 & 0 \\ 0 & 0 & 5 & 6 \\ 0 & 0 & 5 & 6 \end{bmatrix}$$

Similarly, column cluster average and expanded column cluster average can be obtained by $B_c = AC$ (proper normalization necessary to get the column centroids) and $C_c = ACC^T$ as shown below.

$$B_c = \begin{bmatrix} 1.5 & 0 \\ 7.5 & 0 \\ 0 & 4.5 \\ 0 & 6.5 \end{bmatrix}$$

$$C_c = \begin{bmatrix} 1.5 & 1.5 & 0 & 0 \\ 7.5 & 7.5 & 0 & 0 \\ 0 & 0 & 4.5 & 4.5 \\ 0 & 0 & 6.5 & 6.5 \end{bmatrix}$$

Finally, compressed co-cluster average and expanded co-cluster average can be calculated respectively by $C_w = R^T AC$ (proper normalization necessary to get the co-cluster centroid matrix) and $C_w = RR^T ACC^T$ as follows.

$$B_w = \begin{bmatrix} 4.5 & 0 \\ 0 & 5.5 \end{bmatrix}$$

$$C_w = \begin{bmatrix} 4.5 & 4.5 & 0 & 0 \\ 4.5 & 4.5 & 0 & 0 \\ 0 & 0 & 5.5 & 5.5 \\ 0 & 0 & 5.5 & 5.5 \end{bmatrix}$$

In summary $\left\|\mathrm{A} - \hat{\mathrm{A}}\right\|_2^2 = \|\mathrm{A} - RR^T ACC^T\|_2^2 = \|\mathrm{A} - C_w\|$. Note: Objective function of usual k-means algorithms can be represented by $Q\left(\{\pi\}_{j=1}^k\right) = \sum_{j=1}^k \sum_{x \in \pi_j} \left\|x - C_j\right\|_2^2 = \left\|\mathrm{A} - \hat{\mathrm{A}}\right\|_2^2$, where $\hat{\mathrm{A}} = RR^T A$.

Cho et al. (2014) developed an online co-clustering algorithm that updates both row and column assignment simultaneously. The algorithm was proposed to handle the huge storage requirement for large-scale data as well as the incremental availability of certain data (i.e., online transaction data. The biggest advantage is that it does not keep all the data in the main memory. However, the ordered data is processed one by one and only updates the affected data's statistics. In this way, the streaming data can be managed successfully.

Spectral co-clustering (Dhillon, 2001) brings a novel approach to the bipartite graph partitioning problem. Before the spectral co-clustering, the existing algorithms were not able to cluster both documents and words. The proposed Spectral co-clustering modeled document collection as a bipartite graph to cluster both documents and words simultaneously. It was shown that the second left and right most singular matrix of a word-document matrix is an optimal solution to a real relaxation in bipartition problem. Below is the outline of spectral co-clustering.

Table 2

Spectral Co-Clustering

Given A $\in \mathrm{R}^{m\,x\,n}$

[RC, CC] = **spectralCoclustering**(A, k)

(1) $A_n = D_1^{-1/2} \mathrm{A}\, D_2^{-1/2}$,

where D_1 = diag(sum(A,2))

D_2 = diag(sum(A,1))

(2) U, Σ, V^T = SVD(A_n) where,

$U_K \in \mathrm{R}^{m\,x\,k}, \Sigma \in \mathrm{R}^{k\,x\,k}, V_k^T \in \mathrm{R}^{k\,x\,n}$

(3) $\mathrm{E} = \begin{bmatrix} D_1^{-1/2} U_{1:l} \\ D_2^{-1/2} V_{1:l} \end{bmatrix} \in \mathrm{R}^{(m+n)\,x\,l}$,

where $l = ceil(log_2(k))$.

(4) [RC, CC] = k-means(E, k)

First, the data matrix is normalized by both the square-root of its row sum and the square-root of its column sum. Secondly, the left singular vector matrix (U), the eigenvalue matrix (Σ), and the right singular vector matrix (V) are obtained using singular value decomposition of the normalized matrix. Thirdly, the left singular vectors and the right singular vectors, again normalized by the square-root of its row sum and the square-root of its column sum, respectively, are concatenated. Finally, the usual Euclidean k-

Means clustering is applied to get both row and column clustering assignments with the resulting concatenated singular vector matrix as an input data.

Following table summarizes the differences between the traditional K-means, Spherical K-means and Stochastic K-means clustering algorithms.

Table 3

K-means vs Spherical K-means vs Stochastic K-means

	Clustering	Normalization	Metrics
K-means	Row (or Column)	N/A	Euclidean
Spherical K-means	Row (or Column)	2-norm	Cosine
Stochastic K-means	Row (or Column)	1-norm	KL-divergence

The stochastic co-clustering and spherical co-clustering is a brand new approach which uses an original k-means framework, feature construction, and different normalization techniques.

In the upcoming section, we show the methods used during the development of the co-clustering algorithm and provide two practical examples of the algorithms.

CHAPTER III

Proposed Work

Feature Construction

Feature construction is creation of additional features that discovers missing information. Stochastic co-clustering uses feature construction to add additional features to the data matrix. Given data matrix of m rows and n columns, where $m \geq n$, we append *m-n* one's matrix to the data matrix horizontally (Motoda, and Liu, 2002). This process allows us to apply Sinkhorn-Knopp (SK) Normalization, which requires input data matrix to be square. After the SK normalization, the sum of each row and each column of data matrix is equal to 1 (Sinkhorn and Knopp, 1967).

Normalization

Raw data, which is an original form of the collected data, do not disclose the deviation from the central tendency (Cho, 2008). Thus, in many aspects of the Data Mining, it is the most crucial steps because the variation in the dataset will give an insight about the data. Data transformation is the process to transform the format of the data or its structure. Livne and Golub (2004) offer an iterative algorithm, called BIN for a square matrix and NBIN for a rectangular matrix, which scales all the rows and columns of a matrix to have L2-norm of the unit. Bi-normalization is one of the data transformation techniques which we chose to apply to proposed algorithm. The results for bi-normalization of a rectangular matrix as follow (Livne and Golub, 2004).

$$\sum_{j=1}^{n} a'^{2}_{ij} = n \; for \; j = 1, \dots, m$$

and

$$\sum_{i=1}^{m} a'^{2}_{ij} = m \; for \; i = 1, \ldots, n$$

The usefulness of the bi-normalization on the co-clustering algorithm has been shown by Cho et al. They have implemented the MSSRCC algorithm which 6 different transformation techniques and obtained sufficient accuracy with NBIN normalization.

The performance with different data transformation techniques was compared in Figure 4 (Cho et al. 2008). To be more specific, the following strategies used for performance comparison. (1) no transition (NT), row/column standardization (RS/CS), double centering (DC) and bi-normalization (NBIN). The good performance MSSRCC with NBIN is encouraging as we target to embed NBIN into the proposed algorithm steps to achieve bi-spherical normalization.

Table 4

Pseudocode of Spherical Co-Clustering Algorithm

1. Horizontal extension of a given data matrix with one's matrix.
2. Apply bi-normalization
3. Vertical Concatenation of an output of bi-normalization with its transpose
4. Apply k-means clustering algorithm using Cosine similarity as a distance metric

Sinkhorn-Knopp(SK) is another data transformation technique where it takes a square matrix A and finds Diagonal Matrices D1 and D2 so that D1 A D2 is a doubly stochastic, i.e., the sum of each row and column is equal to 1 (Sinkhorn et al., 1964). In the stochastic co-clustering algorithm, we apply SK normalization after feature construction, i.e., the horizontal extension of a given dataset.

$$\sum_i a_{ij} \; = \sum_j a_{ij} = 1$$

As shown in the algorithm definition, the output after SK normalization, the sum of each row and column is equal to one.

Table 5

Pseudocode of Stochastic Co-Clustering Algorithm

1. Horizontal extension of a given data matrix with one's matrix.
2. Apply SK Normalization
3. Vertical Concatenation of an output of SK normalization with its transpose
4. Apply k-means clustering algorithm using KL divergence to the final matrix

In general, the clustering algorithm can be divided into three major sections. The first section is the Data Preparation where it is crucial to receive desired success. In this step, we modify/normalize the dataset according to needs. In our case, we did not make any modification on an Iris Dataset. The second section is initialization which we have experienced the importance during the application process. We initially used random initialization for k-means clustering. Then, we changed the initialization techniques to the Spectral initialization.

Stochastic and Spherical Co-clustering

Given data matrix A where $A \in R^{m \times n}$ where $m \geq n$, whose i, j element is devoted by a_{ij} is defined as follows:

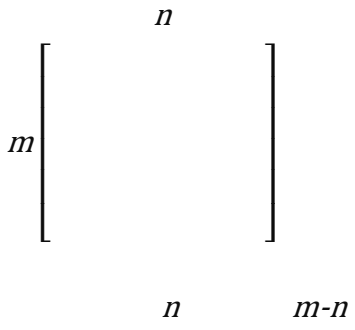
$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

For example, for an Iris dataset can be defined as a data matrix. In that case, rows of the data matrix will be instances, and column of the data matrix would be features.
In this section, we explain each step of the both Stochastic and Spherical Co-clustering in depth.

Feature construction is essential part in data mining where its purpose is to covert data matrix into a desired representation to make it work with the algorithm (Brank et al., 2011). The first step is a feature construction where we concatenate one's matrix to the original data matrix. If a constructed data matrix is not a square matrix, its required to apply a feature construction, i.e., adding new features to the data matrix, to make it square. We implement feature construction to form a square matrix where it will be plausible to extend the data matrix with its transpose. Here is an example of a feature construction.

$A \in R^{m \times n}$, where $m \geq n$

- Given data matrix, A:



- Extended Column, A': $m \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$

As it can be seen, extended data matrix, A' , is now ready for a normalization. Second step in a stochastic co-clustering, we use SK normalization where it normalizes each row and column of a data matrix to L1 norm unit. As a result of SK normalization, sum of each row and column of a data matrix is equal to one (Sinkhorn and Knopp, 1967).
We use bi-normalization, i.e. NBIN, where it normalized each row and column of a data matrix to L2 norm unit. The purpose behind the normalization is to scale each attribute in a range of 0 and 1.

Traditional co-clustering algorithms uses complex scheme behind the scene. In this study, our primary goal is to provide a simple co-clustering algorithm where it uses traditional k-means clustering framework without developing complicating traditional co-clustering algorithm steps. For this purpose, we are extending the normalized square matrix with its transpose.

Extended and normalized data matrix, E where $E \in R^{m \times m}$ is extending by its transpose vertically $E; E^{\mathrm{T}}$. The resulted matrix, E' $where\ E' \in R^{2m \times m}$ is now ready for a traditional one-way clustering.

$$B \in R^{2m \times m}$$

$$m$$

$$m\begin{bmatrix} & \\ & \\ & \end{bmatrix}$$

$$m\begin{bmatrix} & \\ & \\ & \end{bmatrix}$$

When we apply one-way clustering, i.e., either row or column clustering, we would actually apply co-clustering to the data matrix. Since we extended the original data matrix, resulted matrix B, is a result of both instances and features of a data matrix.

In what follows, we provide the pseudocode for each of the proposed stochastic and the spherical co-clustering algorithms with a toy example.

Algorithm 1: Stochastic Co-clustering

Stochastic Co-clustering (A, k)
input: Data matrix A $\in R^{m\times n}$ and cluster number k, where we assume $m \geq n$

Given data matrix: $A = \begin{matrix} \\ m \end{matrix}\begin{matrix} n \\ \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \end{matrix}$

1) Extend Column: $A' = \begin{matrix} \\ m \end{matrix}\begin{matrix} n & m\text{-}n \\ \begin{bmatrix} 1 & 0 & \mathbf{1} \\ 1 & 1 & \mathbf{1} \\ 0 & 1 & \mathbf{1} \end{bmatrix} \end{matrix}$

2) Apply bi-stochastic normalization using the Sinkhorn-Knopp's algorithm (Sinkhorn and Knopp, 1967):

$$E = \begin{bmatrix} 0.61802575 & 0.38196286 & 0 \\ 0 & 0.38196286 & 0.61802575 \\ 0.38197425 & 0.23607427 & 0.38197425 \end{bmatrix},$$

where $\sum A_{i.} = \sum A_{.j} = 1,\ 1 \le i, j \le m$

3) Concatenate normalized matrices:

B = (E; E'^{T}), which results in

$$\begin{array}{c} \\ m \\ m \end{array} \begin{array}{c} \begin{matrix} m & & \end{matrix} \\ \begin{bmatrix} 0.61802575 & 0.38196286 & 0 \\ 0 & 0.38196286 & 0.61802575 \\ 0.38197425 & 0.23607427 & 0.38197425 \end{bmatrix} \\ \begin{bmatrix} 0.61802575 & 0 & 0.38197425 \\ 0.38196286 & 0.38196286 & 0.23607427 \\ 0 & 0.61802575 & 0.38197425 \end{bmatrix} \end{array}$$

4) Apply K-means clustering with KL divergence to B, where B $\in R^{2m \times m}$.

Algorithm 2: Spherical Co-clustering

Stochastic Co-clustering (A, k)
input: Data matrix A $\in R^{m\times n}$ and cluster number k, where we assume $m \geq n$

Given data matrix $A = \begin{matrix} \\ m \end{matrix} \begin{matrix} \overset{n}{} \\ \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \end{matrix}$

1) Extended Column: $A' = \begin{matrix} \\ m \end{matrix} \begin{matrix} \overset{n \quad m\text{-}n}{} \\ \begin{bmatrix} 1 & 0 & \mathbf{1} \\ 1 & 1 & \mathbf{1} \\ 0 & 1 & \mathbf{1} \end{bmatrix} \end{matrix}$

2) Apply bi-normalization using the NBIN((Livne and Golub, 2004):

$$E = \begin{bmatrix} 1.3617 & 0 & 1.0705 \\ 1.0705 & 1.0705 & 0.8415 \\ 0 & 1.3617 & 1.0705 \end{bmatrix},$$

where $\|A_{i.}\| = \|A_{.j}\| = \sqrt{m} \quad 1 \leq i, j \leq m$

3) Concatenerate normalizaed matrices:

B = (E; E'^{T}) , which results in

$$\begin{matrix} \\ m \\ m \end{matrix} \begin{matrix} \overset{m}{} \\ \begin{bmatrix} 1.3617 & 0 & 1.0705 \\ 1.0705 & 1.0705 & 0.8415 \\ 0 & 1.3617 & 1.0705 \end{bmatrix} \\ \begin{bmatrix} 1.3617 & 1.0705 & 0 \\ 0 & 1.0705 & 1.3617 \\ 1.0705 & 0.8415 & 1.0705 \end{bmatrix} \end{matrix}$$

4) Apply K-means clustering using Cosine similarity to B, where B $\in$ R $^{2m \text{ x } m}$

In summary, we have used four different normalization (including one-norm, two-norm, SK and nbin), data extension, and three different metrics (including Euclidean distance, cosine similarity and KL divergence). Table 7 summarizes all the mentioned algorithm strategies.

Table 6

Summarization of the Clustering Algorithms

Algorithm	Normalization	Data Extension	Distance Metric
K-means Clustering	N/A	N/A	Euclidean Distance
Spherical Clustering	Cosine Normalization	N/A	Cosine Similarity
Stochastic Clustering	1-norm	N/A	KL Divergence
Spectral Co-clustering	Square-root of 1-norm	Yes	Euclidean Distance
Stochastic Co-Clustering	SK Normalization	Yes	KL Divergence
Spherical Co-Clustering	Bi-normalization	Yes	Cosine Similarity

Note. Summarization of the Algorithms we have studied in our research

CHAPTER IV

Experiments

Development and Experiment Environments

The six different clustering algorithms, including the existing four algorithms (k-means with Euclidean distance, Cosine similarity, and KL-divergence, respectively, as well as spectral co-clustering) and the two proposed algorithms (stochastic co-clustering and spherical co-clustering).

Python script language (version 3.7) on a Spyder IDE (version 3.3.1) was used as the main development and experiment environment. Furthermore, Bi-normalization was implemented in MATLAB. However, SK normalization package available online and installed using package-management system(PIP) normalize the matrix bi-stochastically.

All the algorithm development and the experiments were conducted on Windows 10 operation system.

Dataset

In this study, we used two different datasets to apply proposed algorithms. Both datasets are publicly available for academic purposes.

1) Iris Dataset

Iris is one of the famous datasets created by Ronald Fisher in his research. Iris dataset contains 150 instances and 5 features where the features are petal length, petal width, sepal length, sepal width and species respectively. We applied 5 different algorithms to Iris dataset and created a model accuracy table (Fisher, 1936)

2) Seeds Dataset

Dataset contains geometrical properties of three different varieties of wheat: Kama, Rosa and Canadian. Seed dataset contains 210 instances and 7 features where

features are area, perimeter, compactness, length and width of kernel, asymmetry coefficient, and length of kernel groove respectively (Charytanowicz et al., 2010). Visualization of the kernel structure obtained by the usage of X-ray techniques.

Cluster Label Initialization

K-means clustering does not guarantee identical clusters because random initialization of the data points leads to a different cluster each time. Hence, it is a crucial problem in DM and ML to choose stable initialization (i.e. seed) for data points.

Through our experiment, we have used two different initialization method for the initial label assignment. As we already explained in section 2, there are a couple of ways to initialize the cluster label for k-means clustering. The first initialization technique we used was random initialization, where we assigned the cluster label to each row (or column) randomly. However, we did not receive the expected efficiency from it due to the local minimum. K-means clustering algorithm's objective is to minimize the distance between each data point and cluster centroid, however; bad initial seed assignment leads to local minimum trap (Khan and Ahmad, 2004).

As we could not get the expected results, we have decided to use spectral initialization. Spectral co-clustering, which denoted in table 2, results in row cluster (RC) and column cluster (CC). When we use RC to initialize the cluster label for data points, the accuracy of the clusters has been improved, as it is more stable then k-means clustering (Meila, 2015).

Figure 1. Summarization of major experimental steps.

CHAPTER V

Results & Discussion

In this chapter, we present empirical results that shows the usefulness of the created algorithms. We perform 5 different clustering algorithms (Euclidean k-means, spherical k-means, stochastic k-means, spherical co-clustering, stochastic co-clustering) to the both Iris and Seed datasets. Each algorithm is summarized in table 2.

As we have mentioned in the Chapter 1, the objective function is a metrics that the applied algorithm wants to minimize or maximize according to the metric used.

When we use Euclidean as a distance metric for the k-means clustering, we expect it to monotonically decrease. When we use cosine as a distance metric spherical clustering, we expect it to increase. The monotonicity of the cosine similarity has been proved in Spherical K-means Clustering (Dhillon and Modha, 2001). We applied 5 algorithms with two different cluster label initiation technique: random and spectral. As we mentioned earlier, the KL divergence calculates the divergence between two probably distribution, thus, its objective is to minimize it. The convergence of the algorithm has been proved in Clustering with Bregman Divergences (Banerjee et al., 2004).

Random Cluster Label Initialization

Following results has been generated using random initialization for each data points.

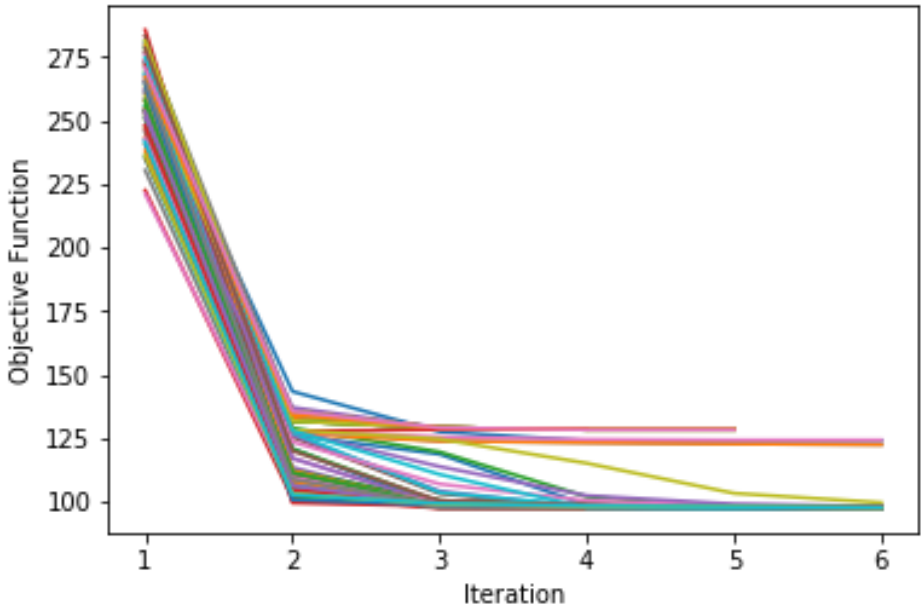


Figure 2. Objective function value change over an iteration.

The above figure generated using k-means clustering on an iris dataset for 100 iteration. Since we have used random initialization each time, the objective function shows versatality.

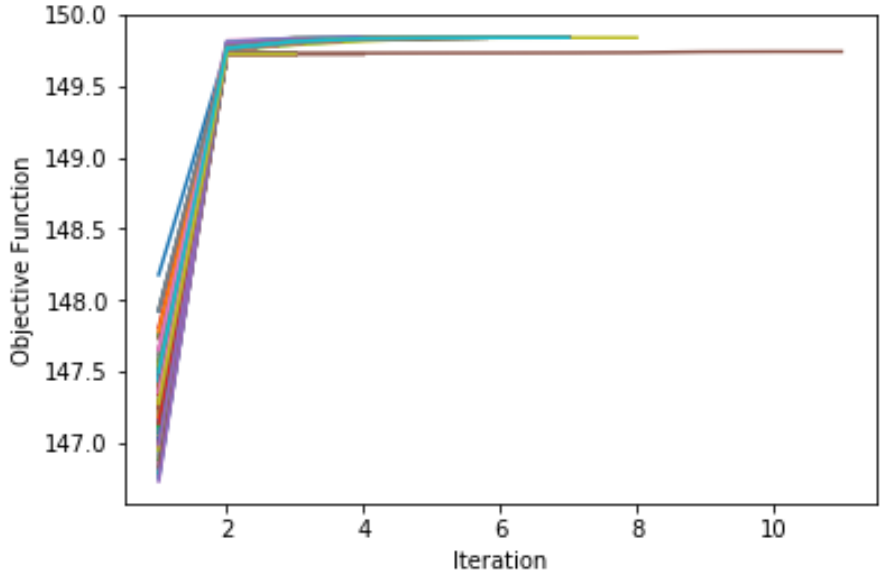


Figure 3. Objective function change over iteration for Spherical Clustering

The above figure generated using spherical clustering on an iris dataset for 100 iteration. Since we have used random initialization each time, the objective function shows versatality.

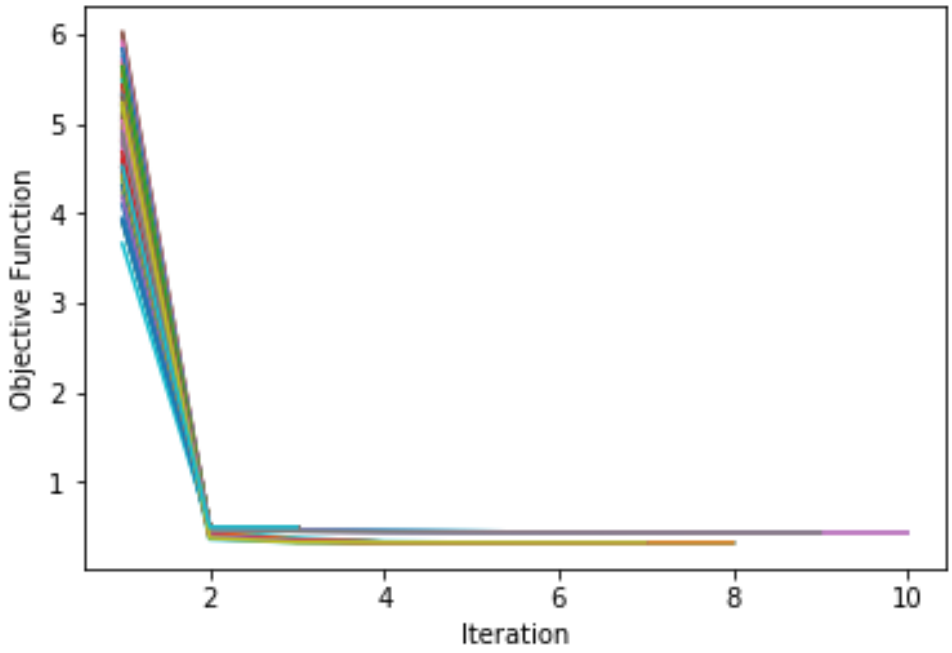


Figure 4. Objective function change over iteration for Stochastic Clustering.

The above figure generated using stochastic clustering on an iris dataset for 100 iteration. Since we have used random initialization each time, the objective function shows versatality.

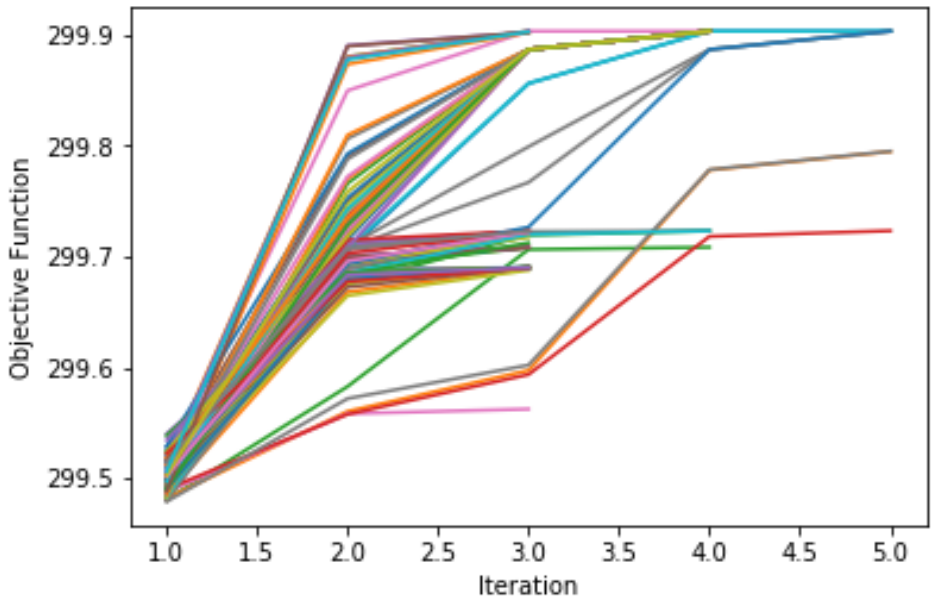


Figure 5. Objective function change over iteration for Spherical Co-Clustering. The above figure generated using Spherical Co-clustering on an iris dataset for 100 iteration.

Since we have used random initialization each time, the objective function shows versatality.

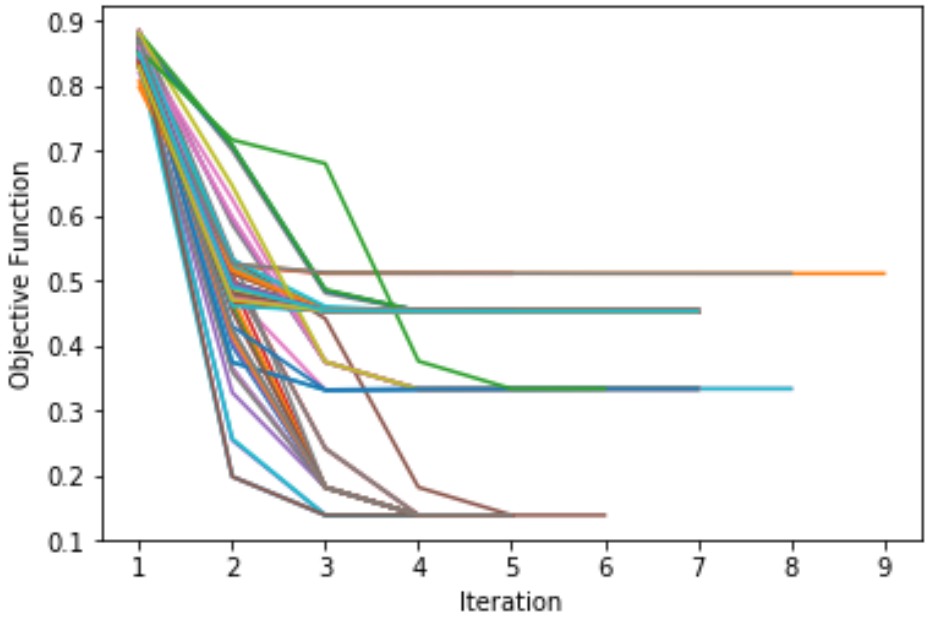


Figure 6. Objective function change over iteration for Spherical Co-Clustering. The above figure generated using Stochastic Co-clustering on an iris dataset for 100 iteration. Since we have used random initialization each time, the objective function shows versatality.

As it can be seen from the above objective function change over iteration, the monotonicity of the both spherical and stochastic co-clustering algorithms are validated.

Accuracy

The step after applying the algorithm to the given dataset is a performance evaluation of the model. Although there are different types of metrics to achieve the performance, we will use the accuracy metrics. The confusion matrix itself is not a performance measurement metric (medium.com, 2017), but the accuracy metric is calculated using the number from it. The confusion matrix is a table with two and columns that summarizes the number of false positive (FP), false negative (FN), true positive (TP) and true negative (TN). The accuracy value can be calculated using the following formula:

$$accuracy\ (\%) = \frac{TP+TN}{TP+TN+FP+FN} \times 100.$$

As all the algorithms include the randomness in cluster initialization, we compute the mean accuracy for every algorithm over 100 random runs. Note that spectral co-clustering algorithms also generate different clustering result as it utilizes the usual Euclidean k-means with random initialization. We first form a confusion matrix of the total samples and then maximize the diagonal of the confusion matrix because cluster labels are permuted.

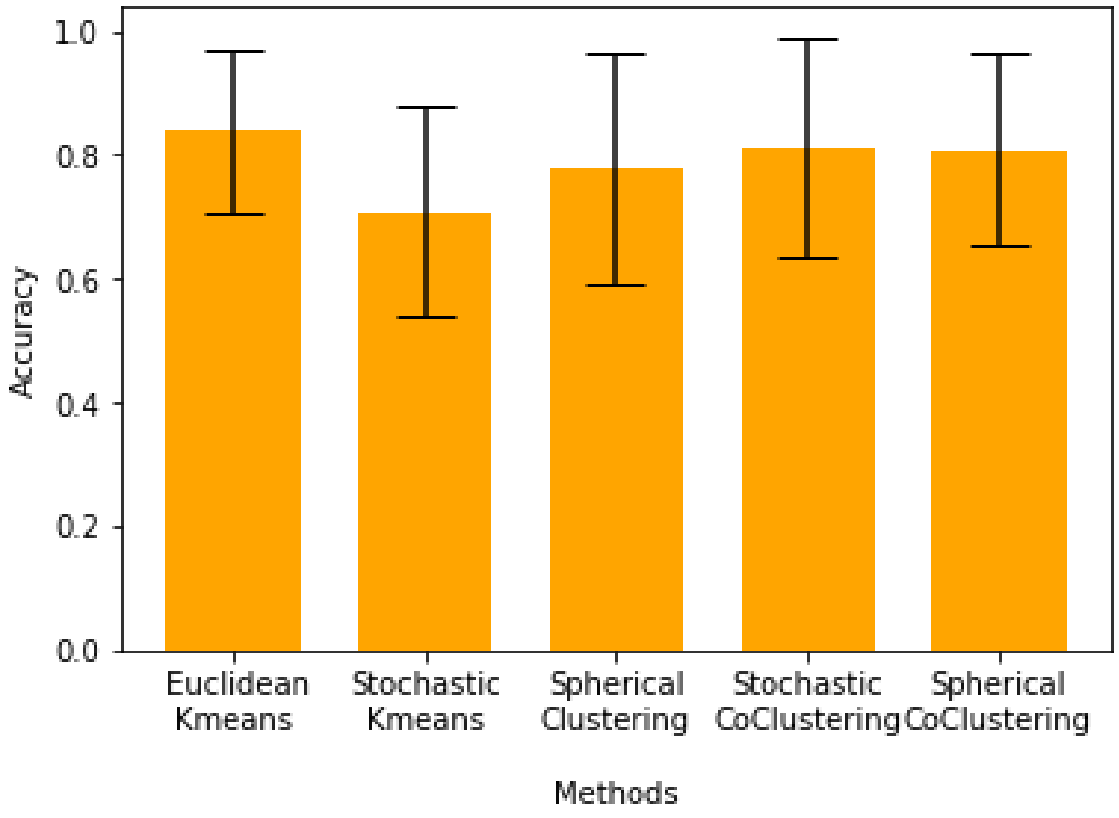


Figure 7. Comparison of 5 approach over accuracy mean and std deviation.

Spectral Initialization

We have applied same algorithms using the output of Spectral Co-clustering (RC), as an initial seed. Following figures has been generated:

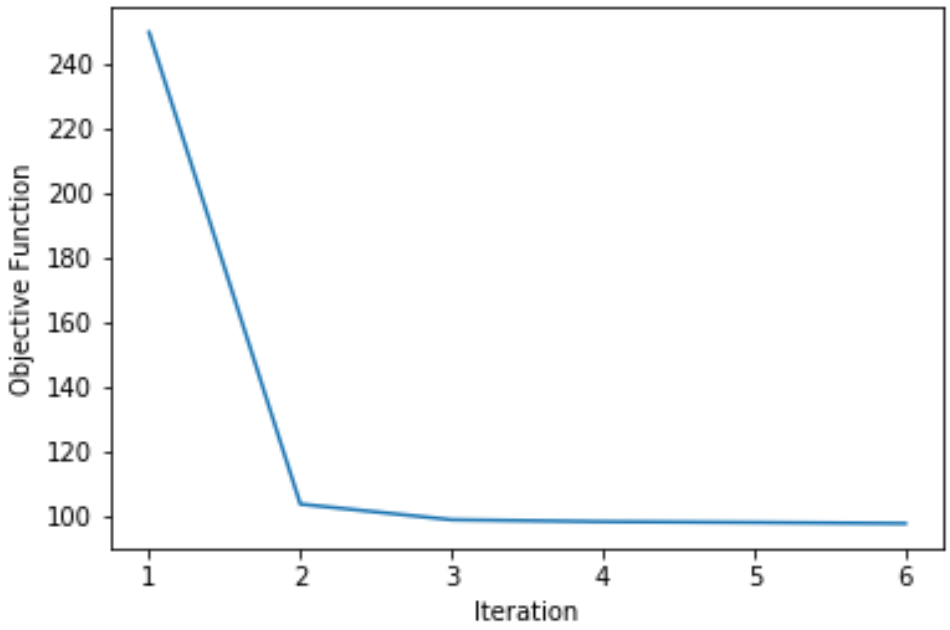


Figure 8. Objective function value change over an iteration.

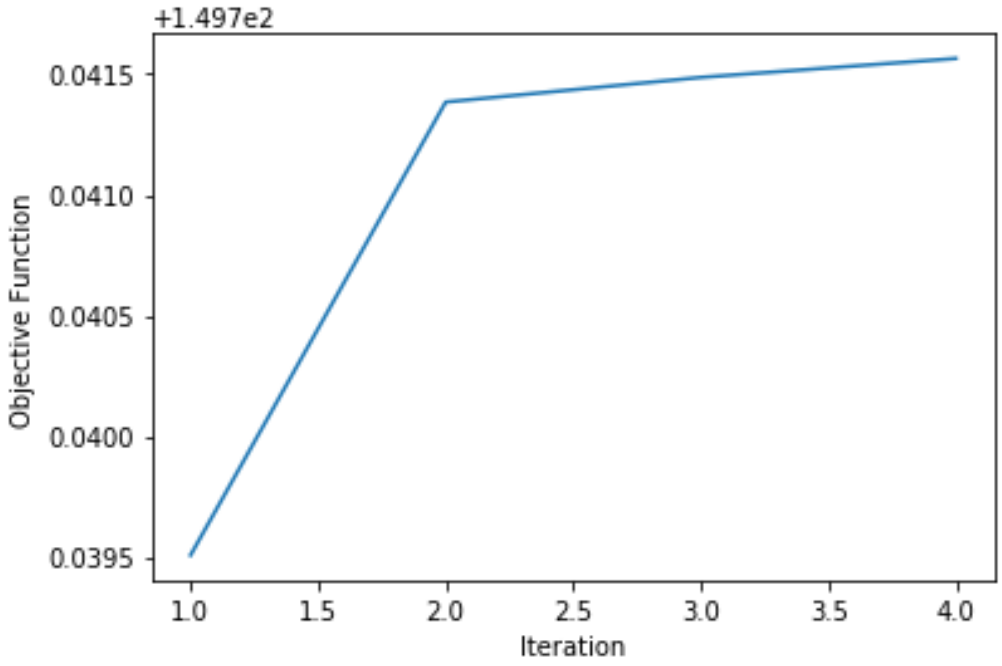


Figure 9. Objective function value change over an iteration.
The above figure generated using spherical clustering on an iris dataset for an iteration

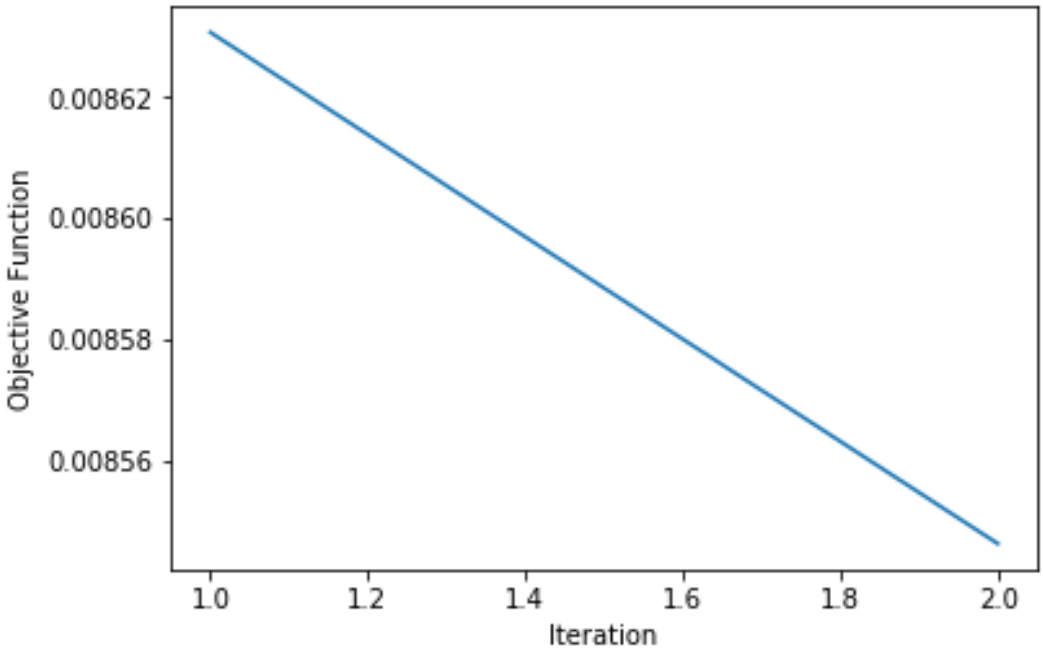


Figure 10. Objective function value change over an iteration.
The above figure generated using stochastic clustering on an iris dataset for an iteration

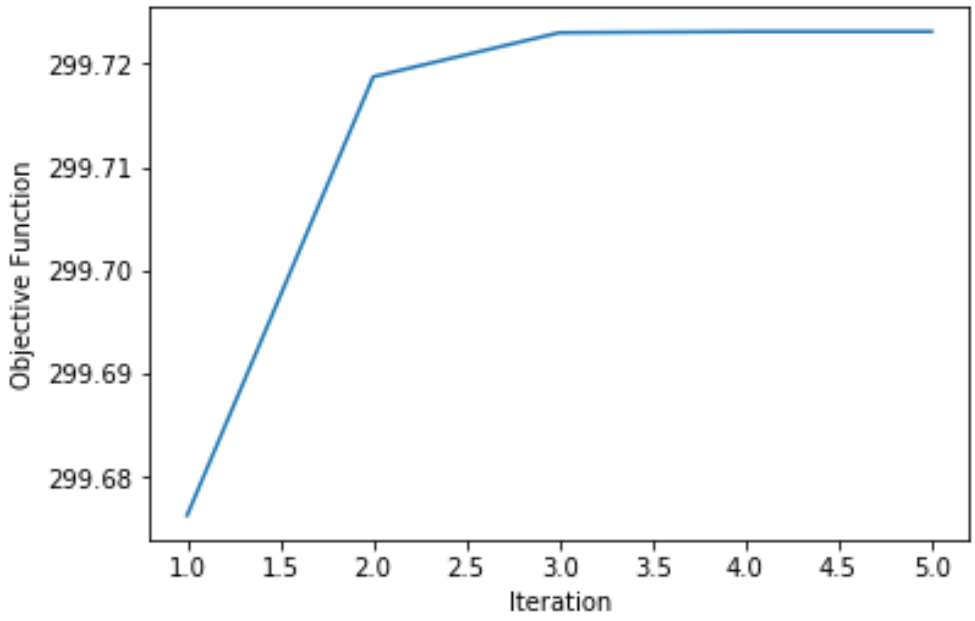


Figure 11. Objective function value change over an iteration.
The above figure generated using spherical co-clustering on an iris dataset for an iteration

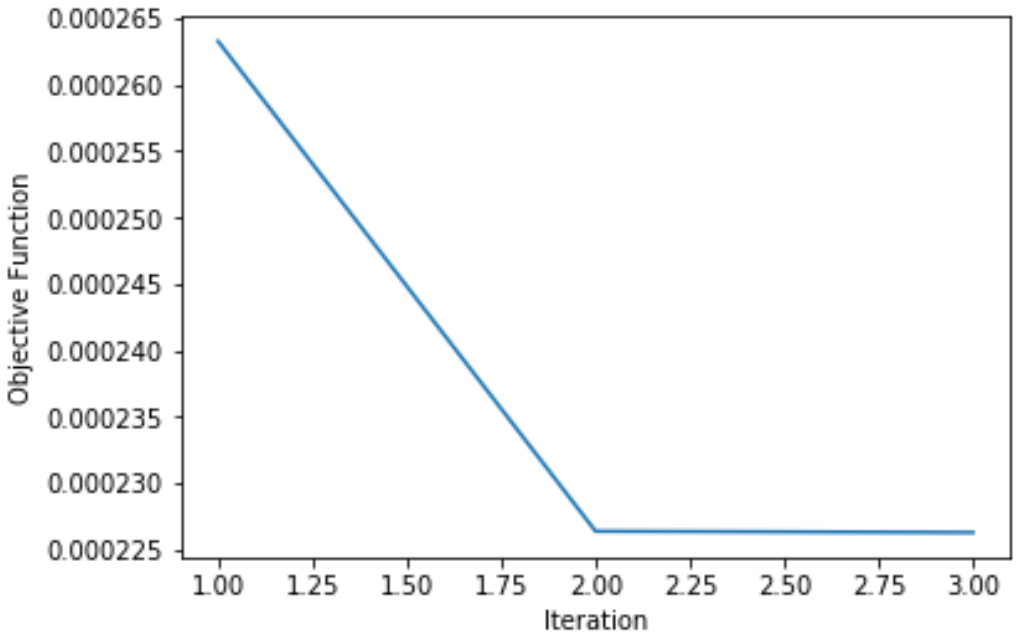


Figure 12. Objective function value change over an iteration.
The above figure generated using stochastic co-clustering on an iris dataset for an iteration

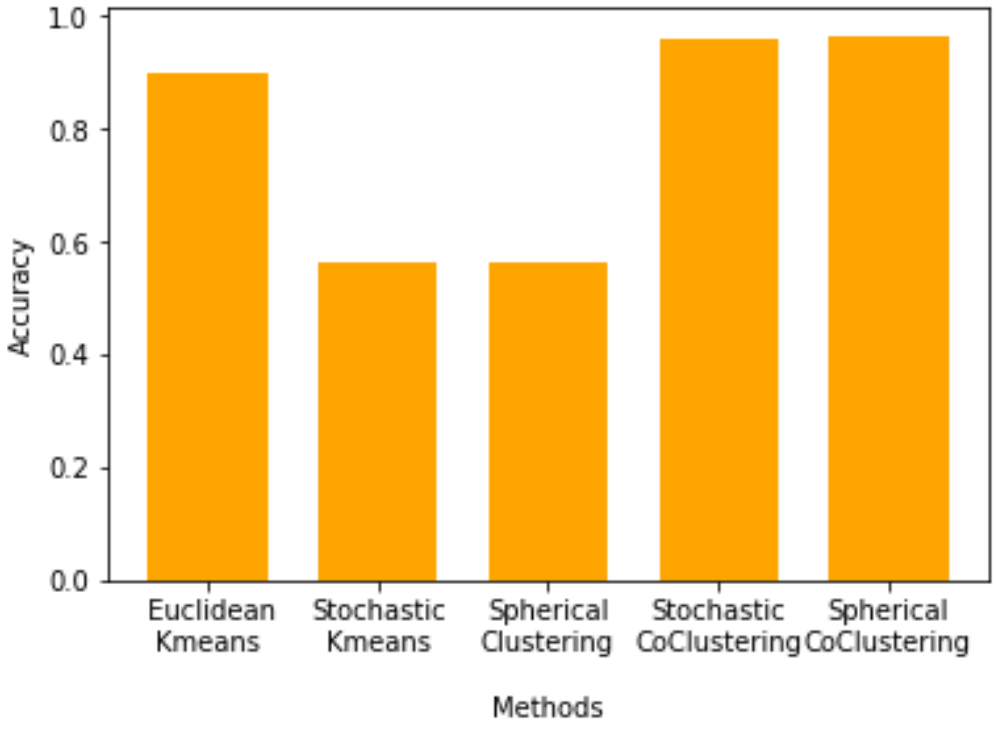


Figure 13. Comparison of 5 approach over accuracy.

Conclusion

In this study, we have created two different co-clustering algorithms, stochastic and spherical co-clustering, which uses unique data transformation, normalization techniques, and traditional k-means clustering, to cluster both instance and feature of a data matrix simultaneously. Our main objective in this study was to use primary k-means clustering structure instead of a complex traditional co-clustering algorithm.

To summarize, we first applied feature construction (i.e., adding new features), to make the data matrix square. Given data matrix A, where $A \in R^{m \times n}$, we concatenated one's matrix, where $\in R^{m \times m-n}$. Next, we applied specific data normalization techniques to normalize the data matrix. We used bi-normalization and Sinkhorn-Knopp normalization algorithms for Spherical and Stochastic Co-Clustering. Un-normalized data

matrix does not reveal the natural tendency of the data. When we normalize the given data matrix, we set each element to a specific range.

The next step is to concatenate the normalized data matrix with its transpose. In this case, resulted data matrix has both row and column information in it. Finally, we apply a one-way clustering algorithm to cluster row (or column) vise.

Using publicly available dataset, Iris, we applied 5 algorithms (Euclidean k-means, spherical k-means, stochastic k-means, spherical co-clustering, stochastic co-clustering) to compare the accuracy. As we explained earlier, the last step of the proposed algorithms is k-means clustering. K-means clustering requires initial cluster assignment of each data point. Initially, we used a random initialization technique to assign each data point to cluster randomly. Unfortunately, generated algorithms did not show the expected accuracy rate due to local minimum trap. Random initialization of each data point results in different cluster assignment on each iteration thus, each cluster assignment was not identical. To overcome the issue, we used a more stable data initialization technique, spectral initialization, for the label assignment. Spectral initialization on a stochastic co-clustering has shown successful accuracy percentage over a Euclidean k-means clustering.

Finally, we compare the accuracy performance of the five different algorithms using both spectral and random initialization on Iris dataset. Our empirical results show that a stochastic co-clustering shows better accuracy over the traditional one-way k-means clustering.

REFERENCES

I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001), pages 269–274, 2001.

Cho, H.; An, M. K. (2014): Co-clustering algorithm: batch, mini-batch, and online. International Journal of Information & Electronics Engineering, vol. 4, no. 5, pp. 340-346

MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proc. 5th Berkeley Symp. Math. Statist, Prob., vol. 1, pp. 281–297 (1967)

J. A. Hartigan. Direct clustering of a data matrix. Journal of the American Statistical Association, 67(337):123–129, March 1972.

S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: A survey," in Proc. IEEE/ACM TCBB, vol. 1, no. 1, pp. 24–45, 2004

S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. Machine Learning, 42(1):143–175, January 2001.

Salton, G. and M. J. McGill: 1983, Introduction to Modern Retrieval. McGraw-Hill Book Company

Rasmussen, E.: 1992, 'Clustering Algorithms'. In: W. B. Frakes and R. Baeza-Yates (eds.): Information Retrieval: Data Structures and Algorithms. pp. 419–442.

Duda, R. O. and P. E. Hart: 1973, Pattern Classification and Scene Analysis. Wiley.

Y. Cheng and G. M. Church, “Biclustering of expression data,” in Proc. ISMB’00, 2000, pp. 93–103.

O. E. Livne and G. H. Golub. Scaling by binormalization. Numerical Algorithms, 35(1):97–120, 2004

H. Cho, I. S. Dhillon, Y. Guan, and S. Sra, “Minimum sum squared residue based co-clustering of gene expression data,”inProc. SDM’04, 2004, pp. 114–125.

B. Kwon and H. Cho, “Scalable co-clustering algorithms,”ICA3PP’10, C.-H. Hsu et al., ed., pp. 32–43, Part I, LNCS, vol. 6081, 2010.

J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: a novel symbolic representation of time series. Data Mining and Knowledge

MacQueen, J., “Some Methods for Classification and Analysis of Multivariate Observations,” Proc. of rhe Fifh Berkeley Symposium on Math., Srar. andProb., Vol. 1, pp. 281-296, 1967.

Cho, Hyuk & S Dhillon, Inderjit. (2008). Coclustering of Human Cancer Microarrays Using Minimum Sum-Squared Residue Coclustering. IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM. 5. 385-400. 10.1109/TCBB.2007.70268.

G. Salton. Automatic Text Processing. Addison-Wesley, Dhillion, 1989 I. S. Y. Guan, and J. Kogan. Iterative clustering of high dimensional text data augmented by local search. In Proceedings of the 2nd IEEE International Conference on Data Mining (ICDM’02), 2002.

Brank J., Mladenić D., Grobelnik M. (2011) Feature Construction in Text Mining. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA

Sinkhorn, Richard; Knopp, Paul. Concerning nonnegative matrices and doubly stochastic matrices. Pacific J. Math. 21 (1967), no. 2, 343--348. https://projecteuclid.org/euclid.pjm/1102992505

Kullback, S.; Leibler, R.A. (1951). "On information and sufficiency". Annals of Mathematical Statistics. 22 (1): 79–86. doi:10.1214/aoms/1177729694. MR 0039968.

Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar, Introduction to Data Mining.

https://medium.com/greyatom/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b

Khan, Shehroz & Ahmad, Amir. (2004). Cluster center initialization algorithm for K-means clustering. Pattern Recognition Letters. 25. 1293-1302. 10.1016/j.patrec.2004.04.007.

Marina Meila. Spectral Clustering: a Tutorial for the 2010's. Handbook of cluster analysis, page 753, 2015.

K. Jain, Anil. (2010). Data Clustering: 50 Years Beyond K-Means. Pattern Recognition Letters. 31. 651-666. 10.1016/j.patrec.2009.09.011.

Wagstaff, Kiri & Cardie, Claire & Rogers, Seth & Schrödl, Stefan. (2001). Constrained K-means Clustering with Background Knowledge. Proceedings of 18th International Conference on Machine Learning. 577-584.

S Dhillon, I & Mallela, S & Modha, DS. (2003). Information-theoretic co-clustering. KDD. 89-98. 10.1142/9789812795236_0006.

Cover, T. M., Thomas, J. A. (2006). *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience. ISBN: 0471241954

Arthur, David & Vassilvitskii, Sergei. (2007). K-Means++: The Advantages of Careful Seeding. Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms. 8. 1027-1035. 10.1145/1283383.1283494.

H. Cho and I. S. Dhillon. Effect of data transformation on residue. Technical report, 2007.

Zhao, Weizhong & Ma, Huifang & He, Qing. (1970). Parallel K-Means Clustering Based on MapReduce. Cloud computing. 5931. 674-679. 10.1007/978-3-642-10665-1_71.

Motoda, H., Liu, H.: Feature selection, extraction, and construction. Commun. Inst. Inf. Comput. Mach. Taiwan 5(2), 67–72 (2002)

Banerjee et al., 2004 Banerjee, Arindam, Merugu, Srujana, Dhillon, Inderjit, Ghosh, Joydeep. 2004. Clustering with bregman divergences. J. Machine Learn. Res., 234–245.

VITA

Education

M.S in Computing and Information Science August 2017 – Present
Sam Houston State University(SHSU), Huntsville, Texas

B.S. in Computer Science August 2015 – May 2017
Sam Houston State University(SHSU), Huntsville, Texas

Work Experience

Teaching Assistant September 2016 – Present

Sam Houston State University, Huntsville, Texas

- Assisted advanced courses: COSC 4314 (Data Mining) and COSC 5319 (Algorithm Design and Analysis)
- Taught undergraduate courses: COSC 1436/1437 (Programming Fundamentals I/II) and COSC CSTE 1339 (Introduction to Computer)
- Taught more than 100 students in the programming labs and recitation sections
- Graded papers & program code, held office hours and recitation sections, and lectured programming labs

Im-park

June 2018 – August 2018

Android Developer Intern

- Developed an application that mainly focuses on the open source project called "Google Science Journal"
- The application uses phone's sensors to record more than 8 scientific experiments, such as decibel of the sound from the microphone, ambient light measurement, accelerometers and to investigate moments, etc.
- The application helps more 1000 students on their scientific experiment from the world around them.

Research Assistant September 2016 – January 2017

Sam Houston State University, Huntsville, Texas

Acquisition of Browser Artifacts from Android Devices

- Advised by Dr. Cihan Varol
- Web storage implementation and proof of usage on Android mobile phones.
- Aimed to help digital forensic investigators find artifacts in Android phones

Volunteer Work

- Acted as judge in the "Best Robotics Game Day" event at SHSU (29th October 2016)

- Worked as an organizing committee member at the "Sharp Your Future" event organized by IEEE Firat University (20th April 2014)
- Managed three people in the business with an asset of $15,000 and 50% rise on the daily earning

Skills

- **Web Site Design & Development:** HTML, CSS, jQuery, and Bootstrap
- **Database Design and Management:** MYSQL, SQL
- **General Programming:** Python, Java, and MATLAB
- **Other Familiarities:** Django, R, C++, JavaScript, and node.js

Publication

- SARIBOZ, E., VAROL, C. "Acquisition of Browser Artifacts from Android Devices", *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, Volume 7, Issue 2, pp. 175-182, June 2018
- Spherical Co-Clustering and Its Application – under preparation

Affiliation

- Member of IEEE student organization. (September 2013- August 2014)
- Member of SHACS (Sam Houston Association of Computer Scientist. (September 2016 – Present)
- President in TSO (Turkish Student Organization) (October 2016 – Present)

Honors and Awards

- Certificate of completing high school with honor.
- Scholarship form Computer Science department of Sam Houston State University(Fall 2015 – Spring 2019)
- "Acquisition of Browser Artifacts" research project had been awarded $1,200 by CoSET (College of Science& Engineering Technology)
- Sam Houston State University Dean List recipients. (Spring 2016, Fall 2016, Spring 2018)
- President List Recipient recipients. (Spring 2017, Fall 2019)

- Presenter at the “Undergraduate Research Symposium” (Spring 2017)